Contenido

Prefacio a la cuarta edición xvii

PARTE I. **ALGORITMOS Y HERRAMIENTAS DE PROGRAMACIÓN** **1**

Capítulo 1. Introducción a las computadoras y los lenguajes de programación 3
INTRODUCCIÓN 3
1.1. ¿Qué es una computadora? 4
1.1.1. Origen de las computadoras 5
1.1.2. Clasificación de las computadoras 6
1.2. Organización física de una computadora 7
1.2.1. Dispositivos de Entrada/Salida (E/S): periféricos 8
1.2.2. La memoria principal 9
1.2.3. Unidades de medida de memoria 10
1.2.4. El procesador 12
1.2.5. Propuestas para selección de la computadora ideal para aprender programación o para actividades profesionales 14
1.3. Representación de la información en las computadoras 15
1.3.1. Representación de textos 15
1.3.2. Representación de valores numéricos 16
1.3.3. Representación de imágenes 17
1.3.4. Representación de sonidos 18
1.4. Codificación de la información 19
1.4.1. Sistemas de numeración 19
1.5. Dispositivos de almacenamiento secundario (almacenamento masivo) 21
1.5.1. Discos magnéticos 21
1.5.2. Discos ópticos: CD-ROM y DVD 21
1.5.3. Discos y memorias Flash USB 24
1.5.4. Otros dispositivos de Entrada y Salida (E/S) 24
1.6. Conectores de dispositivos de E/S 26
1.6.1. Puertos serie y paralelo 26
1.6.2. USB 27
1.6.3. *Bus IEEE Firewire – 1394* 27
1.7. Redes, Web y Web 2.0 28
1.7.1. Redes P2P, igual-a-igual (*peer-to-peer*, P2P) 29
1.7.2. Aplicaciones de las redes de comunicaciones 29
1.7.3. Módem 30
1.7.4. Internet y la World Wide Web 30
1.8. El *software* (los programas) 32
1.8.1. Software del sistema 32
1.8.2. Software de aplicación 33
1.8.3. Sistema operativo 34
1.8.3.1. Multiprogramación/Multitarea 35
1.8.3.2. Tiempo compartido (múltiples usuarios, *time sharing*) 35
1.8.3.3. Multiproceso 35

1.9. Lenguajes de programación 36
1.9.1. Traductores de lenguaje: el proceso de traducción de un programa 37
1.9.2. La compilación y sus fases 38
1.9.3. Evolución de los lenguajes de programación 39
1.9.4. Paradigmas de programación 40
1.10. Breve historia de los lenguajes de programación 42
RESUMEN 43

Capítulo 2. Metodología de la programación y desarrollo de software 45
INTRODUCCIÓN 45
2.1. Fases en la resolución de problemas 46
2.1.1. Análisis del problema 47
2.1.2. Diseño del algoritmo 48
2.1.3. Herramientas de programación 48
2.1.4. Codificación de un programa 51
2.1.5. Compilación y ejecución de un programa 52
2.1.6. Verificación y depuración de un programa 52
2.1.7. Documentación y mantenimiento 53
2.2. Programación modular 54
2.3. Programación estructurada 54
2.3.1. Datos locales y datos globales 55
2.3.2. Modelado del mundo real 56
2.4. Programación orientada a objetos 56
2.4.1. Propiedades fundamentales de la orientación a objetos 57
2.4.2. Abstracción 57
2.4.3. Encapsulación y ocultación de datos 58
2.4.4. Objetos 59
2.4.5. Clases 61
2.4.6. Generalización y especialización: herencia 61
2.4.7 Reusabilidad 63
2.4.8. Polimorfismo 63
2.5. Concepto y características de algoritmos 64
2.5.1. Características de los algoritmos 65
2.5.2. Diseño del algoritmo 66
2.6. Escritura de algoritmos 68
2.7. Representación gráfica de los algoritmos 69
2.7.1. Pseudocódigo 70
2.7.2. Diagramas de flujo 71
2.7.3. Diagramas de Nassi-Schneiderman (N-S) 80
RESUMEN 81
EJERCICIOS 81

Capítulo 3. Estructura general de un programa 83
INTRODUCCIÓN 83
3.1. Concepto de programa 84
3.2. Partes constitutivas de un programa 84
3.3. Instrucciones y tipos de instrucciones 85
3.3.1. Tipos de instrucciones 85
3.3.2. Instrucciones de asignación 86
3.3.3. Instrucciones de lectura de datos (entrada) 87
3.3.4. Instrucciones de escritura de resultados (salida) 87
3.3.5. Instrucciones de bifurcación 87
3.4. Elementos básicos de un programa 89
3.5. Datos, tipos de datos y operaciones primitivas 89
3.5.1. Datos numéricos 90
3.5.2. Datos lógicos (*booleanos*) 92
3.5.3. Datos tipo carácter y tipo cadena 92

3.6. Constantes y variables ... 92
3.6.1. Declaración de constants y variables ... 94
3.7. Expresiones ... 94
3.7.1. Expresiones aritméticas ... 95
3.7.2. Reglas de prioridad ... 97
3.7.3. Expresiones lógicas (*booleanas*) ... 99
3.8. Funciones internas ... 102
3.9. La operación de asignación ... 104
3.9.1. Asignación aritmética ... 105
3.9.2. Asignación lógica ... 105
3.9.3. Asignación de cadenas de caracteres ... 105
3.9.4. Asignación múltiple ... 105
3.9.5. Conversión de tipo ... 106
3.10. Entrada y salida de información ... 107
3.11. Escritura de algoritmos/programas ... 108
3.11.1. Cabecera del programa o algoritmo ... 108
3.11.2. Declaración de variables ... 108
3.11.3. Declaración de constantes numéricas ... 109
3.11.4. Declaración de constantes y variables carácter ... 109
3.11.5. Comentarios ... 110
3.11.6. Estilo de escritura de algoritmos/programas ... 111
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS ... 113
CONCEPTOS CLAVE ... 124
RESUMEN ... 124
EJERCICIOS ... 125

Capítulo 4. Flujo de control I: Estructuras selectivas ... 127
INTRODUCCIÓN ... 127
4.1. El flujo de control de un programa ... 128
4.2. Estructura secuencial ... 128
4.3. Estructuras selectivas ... 130
4.4. Alternativa simple (**`si-entonces/if-then`**) ... 131
4.4.1. Alternativa doble (**`si-entonces-sino/if-then-else`**) ... 132
4.5. Alternativa múltiple (**`según_sea, caso de/case`**) ... 137
4.6. Estructuras de decisión anidadas (en escalera) ... 144
4.7. La sentencia **`ir-a`** (**`goto`**) ... 148
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS ... 151
CONCEPTOS CLAVE ... 154
RESUMEN ... 154
EJERCICIOS ... 155

Capítulo 5. Flujo de control II: Estructuras repetitivas ... 157
INTRODUCCIÓN ... 157
5.1. Estructuras repetitivas ... 158
5.2. Estructura **`mientras ("while")`** ... 160
5.2.1. Ejecución de un bucle cero veces ... 162
5.2.2. Bucles infinitos ... 163
5.2.3. Terminación de bucles con datos de entrada ... 163
5.3. Estructura **`hacer-mientras ("do-while")`** ... 165
5.4. Diferencias entre **`mientras`** (**`while`**) y **`hacer-mientras`** (**`do-while`**): una aplicación en C++ ... 167
5.5. Estructura **`repetir`** (**`"repeat"`**) ... 168
5.6. Estructura **`desde/para`** (**`"for"`**) ... 171
5.6.1. Otras representaciones de estructuras repetitivas **`desde/para`** (**`for`**) ... 171
5.6.2. Realización de una estructura **`desde`** con estructura **`mientras`** ... 174
5.7. Salidas internas de los bucles ... 175
5.8. Sentencias de salto **`interrumpir`** (**`break`**) y **`continuar`** (**`continue`**) ... 176
5.8.1. Sentencia **`interrumpir`** (**`break`**) ... 176
5.8.2. Sentencia **`continuar`** (**`continue`**) ... 177

5.9. Comparación de bucles **`while`**, **`for`** y **`do-while`**: una aplicación en C++ 178
5.10. Diseño de bucles (lazos) 179
5.10.1. Bucles para diseño de sumas y productos 179
5.10.2. Fin de un bucle 179
5.11. Estructuras repetitivas anidadas 181
5.11.1. Bucles (lazos) anidados: una aplicación en C++ 183
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS 186
CONCEPTOS CLAVE 197
RESUMEN 197
EJERCICIOS 198
REFERENCIAS BIBLIOGRÁFICAS 199

Capítulo 6. Subprogramas (subalgoritmos): Funciones 201
INTRODUCCIÓN 201
6.1. Introducción a los subalgoritmos o subprogramas 202
6.2. Funciones 203
6.2.1. Declaración de funciones 204
6.2.2. Invocación a las funciones 205
6.3. Procedimientos (subrutinas) 210
6.3.1. Sustitución de argumentos/parámetros 211
6.4. Ámbito: variables locales y globales 215
6.5. Comunicación con subprogramas: paso de parámetros 218
6.5.1. Paso de parámetros 219
6.5.2. Paso por valor 219
6.5.3. Paso por referencia 220
6.5.4. Comparaciones de los métodos de paso de parámetros 221
6.5.5. Síntesis de la transmisión de parámetros 223
6.6. Funciones y procedimientos como parámetros 225
6.7. Los efectos laterales 227
6.7.1. En procedimientos 227
6.7.2. En funciones 228
6.8. Recursión (recursividad) 229
6.9. Funciones en C/C++ , Java y C# 231
6.10. Ámbito (alcance) y almacenamiento en C/C++ y Java 233
6.11. Sobrecarga de funciones en C++ y Java 235
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS 238
CONCEPTOS CLAVE 242
RESUMEN 242
EJERCICIOS 243

PARTE II. **ESTRUCTURA DE DATOS** **245**

Capítulo 7. Estructuras de datos I (arrays y estructuras) 247
INTRODUCCIÓN 247
7.1. Introducción a las estructuras de datos 248
7.2. Arrays (arreglos) unidimensionales: los vectores 248
7.3. Operaciones con vectores 251
7.3.1. Asignación 252
7.3.2. Lectura/escritura de datos 253
7.3.3. Acceso secuencial al vector (recorrido) 253
7.3.4. Actualización de un vector 255
7.4. Arrays de varias dimensiones 258
7.4.1. Arrays bidimensionales (tablas/matrices) 258
7.5. Arrays multidimensionales 260
7.6. Almacenamiento de arrays en memoria 262
7.6.1. Almacenamiento de un vector 262
7.6.2. Almacenamiento de arrays multidimensionales 263

7.7. Estructuras *versus* registros 265
7.7.1. Registros 265
7.8. Arrays de estructuras 266
7.9. Uniones 268
7.9.1. Unión *versus* estructura 268
7.10. Enumeraciones 270
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS 272
CONCEPTOS CLAVE 282
RESUMEN 282
EJERCICIOS 283

Capítulo 8. Las cadenas de caracteres 285
INTRODUCCIÓN 285
8.1. Introducción 286
8.2. El juego de caracteres 286
8.2.1. Código ASCII 286
8.2.2. Código EBCDIC 287
8.2.3. Código universal Unicode para Internet 287
8.2.4. Secuencias de escape 289
8.3. Cadena de caracteres 289
8.4. Datos tipo carácter 291
8.4.1. Constantes 291
8.4.2. Variables 291
8.4.3. Instrucciones básicas con cadenas 292
8.5. Operaciones con cadenas 293
8.5.1. Cálculo de la longitud de una cadena 293
8.5.2. Comparación 294
8.5.3. Concatenación 295
8.5.4. Subcadenas 296
8.5.5. Búsqueda 297
8.6. Otras funciones de cadenas 297
8.6.1. Insertar 298
8.6.2. Borrar 298
8.6.3. Cambiar 299
8.6.4. Conversión de cadenas/números 300
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS 300
CONCEPTOS CLAVE 305
RESUMEN 305
EJERCICIOS 306

Capítulo 9. Archivos (ficheros) 307
INTRODUCCIÓN 307
9.1. Archivos y flujos (stream): La jerarquía de datos 308
9.1.1. Campos 309
9.1.2. Registros 309
9.1.3. Archivos (ficheros) 310
9.1.4. Bases de datos 310
9.1.5. Estructura jerárquica 310
9.1.6. Jerarquía de datos 311
9.2. Conceptos y definiciones = terminología 312
9.2.1. Clave (indicativo) 312
9.2.2. Registro físico o bloque 312
9.2.3. Factor de bloqueo 312
9.3. Soportes secuenciales y direccionables 313
9.4. Organización de archivos 314
9.4.1. Organización secuencial 314
9.4.2. Organización directa 315
9.4.3. Organización secuencial indexada 316

9.5. Operaciones sobre archivos ... 317
9.5.1. Creación de un archivo ... 318
9.5.2. Consulta de un archivo ... 318
9.5.3. Actualización de un archivo ... 319
9.5.4. Clasificación de un archivo ... 319
9.5.5. Reorganización de un archivo ... 320
9.5.6. Destrucción de un archivo ... 320
9.5.7. Reunión, fusión de un archivo ... 320
9.5.8. Rotura/estallido de un archivo ... 321
9.6. Gestión de archivos ... 321
9.6.1. Crear un archivo ... 322
9.6.2. Abrir un archivo ... 322
9.6.3. Cerrar archivos ... 324
9.6.4. Borrar archivos ... 324
9.7. Flujos ... 324
9.7.1. Tipos de flujos ... 325
9.7.2. Flujos en C++ ... 325
9.7.3. Flujos en Java ... 325
9.7.4. Consideraciones prácticas en Java y C# ... 326
9.8. Mantenimiento de archivos ... 326
9.8.1. Operaciones sobre registros ... 328
9.9. Procesamiento de archivos secuenciales (algoritmos) ... 328
9.9.1. Creación ... 328
9.9.2. Consulta ... 329
9.9.3. Actualización ... 332
9.10. Procesamiento de archivos directos (algoritmos) ... 335
9.10.1. Operaciones con archivos directos ... 335
9.10.2. Clave-dirección ... 341
9.10.3. Tratamiento de las colisiones ... 341
9.10.4. Acceso a los archivos directos mediante indexación ... 341
9.11. Procesamiento de archivos secuenciales indexados ... 343
9.12. Tipos de archivos: consideraciones prácticas en C/C++ y Java ... 344
9.12.1. Archivos de texto ... 344
9.12.2. Archivos binarios ... 345
9.12.3. Lectura y escritura de archivos ... 345
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS ... 346
CONCEPTOS CLAVE ... 352
RESUMEN ... 352
EJERCICIOS ... 353

Capítulo 10. Ordenación, búsqueda e intercalación ... 355
INTRODUCCIÓN ... 355
10.1. Introducción ... 356
10.2. Ordenación ... 357
10.2.1. Método de intercambio o de burbuja ... 358
10.2.2. Ordenación por inserción ... 363
10.2.3. Ordenación por selección ... 365
10.2.4. Método de Shell ... 368
10.2.5. Método de ordenación rápida (*quicksort*) ... 370
10.3. Búsqueda ... 374
10.3.1. Búqueda secuencial ... 374
10.3.2. Búqueda binaria ... 379
10.3.3. Búsqueda mediante transformación de claves (*hasting*) ... 383
10.4. Intercalación ... 388
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS ... 391
CONCEPTOS CLAVE ... 402
RESUMEN ... 402
EJERCICIOS ... 403

Capítulo 11. Ordenación, búsqueda y fusión externa (archivos) 405
INTRODUCCIÓN 405
11.1. Introducción 406
11.2. Archivos ordenados 406
11.3. Fusión de archivos 406
11.4. Partición de archivos 410
11.4.1. Clasificación interna 410
11.4.2. Partición por contenido 410
11.4.3. Selección por sustitución 411
11.4.4. Partición por secuencias 413
11.5. Clasificación de archivos 414
11.5.1. Clasificación por mezcla directa 414
11.5.2. Clasificación por mezcla natural 417
11.5.3. Clasificación por mezcla de secuencias equilibradas 421
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS 422
CONCEPTOS CLAVE 426
RESUMEN 426
EJERCICIOS 427

Capítulo 12. Estructuras dinámicas lineales de datos (pilas, colas y listas enlazadas) 429
INTRODUCCIÓN 429
12.1. Introducción a las estructuras de datos 430
12.1.1. Estructuras dinámicas de datos 430
12.2. Listas 431
12.3. Listas enlazadas 433
12.4. Procesamiento de listas enlazadas 436
12.4.1. Implementación de listas enlazadas con punteros 436
12.4.2. Implementación de listas enlazadas con arrays (arreglos) 442
12.5. Listas circulares 450
12.6. Listas doblemente enlazadas 450
12.6.1. Inserción 451
12.6.2. Eliminación 452
12.7. Pilas 452
12.7.1. Aplicaciones de las pilas 458
12.8. Colas 460
12.8.1. Representación de las colas 461
12.8.2. Aprovechamiento de la memoria 467
12.9. Doble cola 468
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS 469
CONCEPTOS CLAVE 476
RESUMEN 477
EJERCICIOS 477

Capítulo 13. Estructuras de datos no lineales (árboles y grafos) 479
INTRODUCCIÓN 479
13.1. Introducción 480
13.2. Árboles 480
13.2.1. Terminología y representación de un árbol general 481
13.3. Árbol binario 482
13.3.1. Terminología de los árboles binarios 483
13.3.2. Árboles binarios completos 484
13.3.3. Conversión de un árbol general en árbol binario 485
13.3.4. Representación de los árboles binarios 489
13.3.5. Recorrido de un árbol binario 493
13.4. Árbol binario de búsqueda 495
13.4.1. Búsqueda de un elemento 497
13.4.2. Insertar un elemento 498
13.4.3. Eliminación de un elemento 499

13.5. Grafos 506
13.5.1. Terminología de grafos 506
13.5.2. Representación de grafos 509
ACTIVIDADES DE PROGRAMACIÓN RESUELTAS 512
CONCEPTOS CLAVE 516
RESUMEN 516
EJERCICIOS 517

Capítulo 14. Recursividad 519
INTRODUCCIÓN 519
14.1. La naturaleza de la recursividad 520
14.2. Recursividad directa e indirecta 524
14.2.1. Recursividad indirecta 527
14.2.2. Condición de terminación de la recursión 528
14.3. Recursión *versus* iteración 528
14.4. Recursión infinita 531
14.5. Resolución de problemas complejos con recursividad 535
14.5.1. Torres de Hanoi 535
14.5.2. Búsqueda binaria recursiva 540
14.5.3. Ordenación rápida (*QuickSort*) 542
14.5.4. Ordenación *mergesort* 545
CONCEPTOS CLAVE 548
RESUMEN 548
EJERCICIOS 549
PROBLEMAS 549

PARTE III. **PROGRAMACIÓN ORIENTADA A OBJETOS Y UML 2.1** **551**

Capítulo 15. Tipos abstractos de datos, objetos y modelado con UML 2.1 553
INTRODUCCIÓN 553
15.1. Programación estructurada (*procedimental*) 554
15.1.1. Limitaciones de la programación estructurada 554
15.1.2. Modelado de objetos del mundo real 555
15.2. Programación orientada a objetos 556
15.2.1. Objetos 557
15.2.2. Tipos abstractos de datos: CLASES 558
15.3. Modelado e identificación de objetos 560
15.4. Propiedades fundamentales de orientación a objetos 561
15.4.1. Abstracción 561
15.4.2. La abstracción en el *software* 561
15.4.3. Encapsulamiento y ocultación de datos 562
15.4.4 Herencia 562
15.4.5. Reutilización o reusabilidad 563
15.4.6. Polimorfismo 564
15.5. Modelado de aplicaciones: UML 565
15.5.1. Lenguaje de modelado 566
15.5.2. ¿Qué es un lenguaje de modelado? 566
15.6. Diseño de software con UML 567
15.6.1. Desarrollo de software orientado a objetos con UML 568
15.6.2. Especificaciones de UML 568
15.7. Historia de UML 568
15.7.1. El futuro de UML 2.1 569
15.8. Terminología de orientación a objetos 569
CONCEPTOS CLAVE 570
RESUMEN 570
EJERCICIOS 570

Capítulo 16. Diseño de clases y objetos: Representaciones gráficas en UML 573
INTRODUCCIÓN 573
16.1. Diseño y representación gráfica de objetos en UML 574
16.1.1. Representación gráfica en UML 575
16.1.2. Características de los objetos 576
16.1.3. Estado 577
16.1.4. Múltiples instancias de un objeto 579
16.1.5. Evolución de un objeto 579
16.1.6. Comportamiento 580
16.1.7. Identidad 582
16.1.8. Los mensajes 582
16.1.9. Responsabilidad y restricciones 584
16.2. Diseño y representación gráfica de clases en UML 584
16.2.1. Representación gráfica de una clase 585
16.2.2. Declaración de una clase 588
16.2.3. Reglas de visibilidad 590
16.2.4. Sintaxis 592
16.3. Declaración de objetos de clases 593
16.3.1. Acceso a miembros de la clase: encapsulamiento 595
16.3.2. Declaración de métodos 597
16.3.3. Tipos de métodos 601
16.4. Constructores 602
16.4.1. Constructor por defecto 603
16.5. Destructores 606
16.6. Implementación de clases en C++ 607
16.6.1. Archivos de cabecera y de clases 608
16.6.2. Clases compuestas 609
16.7. Recolección de basura 610
16.7.1. El método `finalize ()` 610
CONCEPTOS CLAVE 611
RESUMEN 611
EJERCICIOS 613
LECTURAS RECOMENDADAS 614

Capítulo 17. Relaciones entre clases: Delegaciones, asociaciones, agregaciones, herencia 615
INTRODUCCIÓN 615
17.1. Relaciones entre clases 616
17.2. Dependencia 616
17.3. Asociación 617
17.3.1. Multiplicidad 619
17.3.2. Restricciones en asociaciones 620
17.3.3. Asociación cualificada 620
17.3.4. Asociaciones reflexivas 620
17.3.5. Diagrama de objetos 621
17.3.6. Clases de asociación 621
17.3.7. Restricciones en asociaciones 625
17.4. Agregación 626
17.4.1 Composición 628
17.5. Jerarquía de clases: generalización y especialización 629
17.5.1. Jerarquías de generalización/especialización 631
17.6. Herencia: clases derivadas 634
17.6.1. Herencia simple 634
17.6.2. Herencia múltiple 635
17.6.3. Niveles de herencia 636
17.6.4. Declaración de una clase derivada 638
17.6.5. Consideraciones de diseño 639
17.7. Accesibilidad y visibilidad en herencia 640
17.7.1. Herencia pública 640

17.7.2. Herencia privada 640
17.7.3. Herencia protegida 641
17.8. Un caso de estudio especial: herencia múltiple 642
17.8.1. Características de la herencia múltiple 644
17.9. Clases abstractas 645
17.9.1. Operaciones abstractas 646
CONCEPTOS CLAVE 647
RESUMEN 647
EJERCICIOS 648

PARTE IV. METODOLOGÍA DE LA PROGRAMACIÓN Y DESARROLLO DE SOFTWARE **649**

Capítulo 18. Resolución de problemas y desarrollo de software: Metodología de la programación 653
INTRODUCCIÓN 653
18.1. Abstracción y resolución de problemas 654
18.1.1. Descomposición procedimental 654
18.1.2. Diseño descendente 655
18.1.3. Abstracción procedimental 656
18.1.4. Abstracción de datos 656
18.1.5. Ocultación de la información 657
18.1.6. Programación orientada a objetos 657
18.1.7. Diseño orientado a objetos 657
18.2. El ciclo de vida del software 658
18.2.1. El ciclo de vida del software tradicional (*modelo en cascada*) 658
18.2.2. El proceso unificado 660
18.2.3. Cliente, desarrollador y usuario 661
18.3. Fase de análisis: requisitos y especificaciones 662
18.4. Diseño 664
18.5. Implementación (codificación) 666
18.6. Pruebas e integración 666
18.6.1. Verificación 667
18.6.2. Técnicas de pruebas 667
18.7. Mantenimiento 669
18.7.1. La obsolescencia: programas obsoletos 669
18.7.2. Iteración y evolución del software 669
18.8. Principios de diseño de sistemas de software 670
18.8.1. Modularidad mediante diseño descendente 670
18.8.2. Abstracción y encapsulamiento 671
18.8.3. Modificabilidad 671
18.8.4. Comprensibilidad y fiabilidad 672
18.8.5. Interfaces de usuario 672
18.8.6. Programación segura contra fallos 673
18.8.7. Facilidad de uso 673
18.8.8. Eficiencia 674
18.8.9. Estilo de programación, documentación y depuración 674
18.9. Estilo de programación 674
18.9.1. Modularizar un programa en subprogramas 674
18.9.2. Evitar variables globales en subprogramas 675
18.9.3. Usar nombres significativos para identificadores 675
18.9.4. Definir constantes con nombres 676
18.9.5. Evitar el uso de ir (`goto`) 676
18.9.6. Uso adecuado de parámetros valor/variable 676
18.9.7. Uso adecuado de funciones 677
18.9.8. Tratamiento de errores 677
18.9.9. Legibilidad 677
18.10. La documentación 678
18.10.1. Manual del usuario 679

18.10.2. Manual de mantenimiento (documentación para programadores) 680
18.10.3. Reglas de documentación 681
18.11. Depuración 681
18.11.1. Localización y reparación de errores 681
18.11.2. Depuración de sentencias si-entonces-sino 682
18.11.3. Los equipos de programación 683
18.12. Diseño de algoritmos 683
18.13. Pruebas (testing) 684
18.13.1. Errores de sintaxis (de compilación) 685
18.13.2. Errores en tiempo de ejecución 685
18.13.3. Errores lógicos 686
18.13.4. El depurador 686
18.14. Eficiencia 687
18.14.1. Eficiencia *versus* legibilidad (claridad) 689
18.15. Transportabilidad 689
CONCEPTOS CLAVE 689
RESUMEN 690

APÉNDICES **689**

Apéndice A. Especificaciones del lenguaje algorítmico UPSAM 2.0 691

Apéndice B. Prioridad de operadores 713

Apéndice C. Código ASCII y Unicode 717

Apéndice D. Guía de sintaxis del lenguaje C 723

Bibliografía y recursos de programación 751